1. 原型和闭包
2. 对象

* 一切引用类型都是对象，对象是若干属性的集合

1. 对象都是通过函数创建的，函数又是一种对象。
2. 对象实例的\_\_proto\_\_执行对象的prototype

obj1.\_\_proto\_\_===Object.pototype

* 每个函数都有一个prototype，每个对象都有一个\_\_proto\_\_指向创建该对象的函数的prototype
* Object.prototype的\_\_proto\_\_指向null

1. instanceof

instance a of A

沿着a的\_\_proto\_\_这条线找，同时沿着A的prototype这条线找，如果能找到同一个引用则返回。\_\_proto\_这条线即原型链

1. 继承

访问一个对象的属性时，先在自己的属性中找，如果没有，沿着\_\_proto\_\_这条线向上找，这就是原型链。

Object.prototype.hasOwnProperty()自身是否含有某属性

1. 上下文

* 准备工作—语句执行

准备工作：变量、函数表达式—变量声明，默认值为undefined。this—赋值。函数声明—赋值。这3种数据的准备情况我们称为“执行上下文”或“执行上下文环境”。

js在执行一个代码段之前，都会进行这些准备工作来生成执行上下文，这个代码段分3种情况：全局diamond、函数体、eval代码

执行上下文—在执行代码之前，把将要用到的变量都事先拿出来，有值的赋值，没有的赋undefined

* 函数每次调用都产生一个新的执行上下文环境。
* 函数在定义的时候就确定了函数体内部自由变量的作用域。
* this的值是在定义的时候确定的，4种情况

new构造函数 this是构造函数

function Foo(){

this.name="cdx";

console.log(this);//Foo{name:"cdx"}

}

var foo=new Foo();

直接调用构造函数、全局函数、普通函数 this是window

function Foo(){

this.name="cdx";

console.log(this);//window{...}

}

Foo();

function foo2(){

console.log(this);//window{...}

}

foo2();

var foo3=function(){

console.log(this);//window{...}

}

foo3();

函数作为对象的一个属性被调用 this指向该对象

var obj={

x:1,

fn:function(){

console.log(this);//Object{fn:f(),x:1}

}

}

obj.fn();

函数用call apply bind调用 this指向被绑定的对象

在整个原型链中，this值当前对象

function Fn(name){

this.name=name||"cdx",

this.year=1997

}

Fn.prototype.getName=function(){

console.log(this.name);

}

var f1=new Fn();

var f2=new Fn("hi");

f1.getName();//cdx

f2.getName();//hi

1. 作用域

* 作用域最大的用处就是隔离变量，不同作用域可以声明同名变量
* 作用域是在声明时就定义好的，即静态作用域
* 作用域只是一个“地盘”，其中没有变量，要通过作用域对应的执行上下文来获取变量的值。一个作用域中可以有多个执行上下文

1. 闭包

* 2种情况—函数作为返回值，函数做参数

1. other
2. 换行

word-wrap:normal|break-word(截断单词换行)

word-break:normal|break-all(单词都断开)|keep-all(都不断开)

word-break为break-all时，word-wrap失效

1. 同源策略

协议、域名、端口相同

同源策略的限制：

Cookie LocalStorage IndexDB无法获取

DOM无法获得

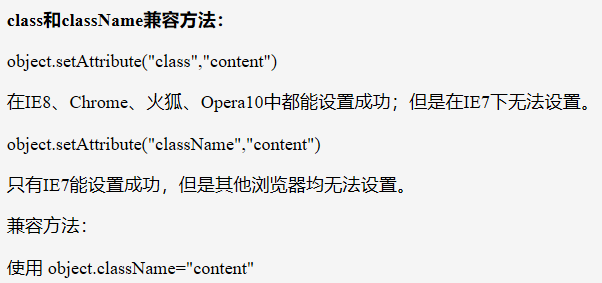
AJAX请求不能发送

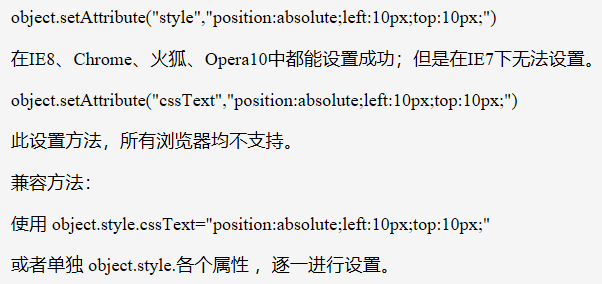
1. 支持度

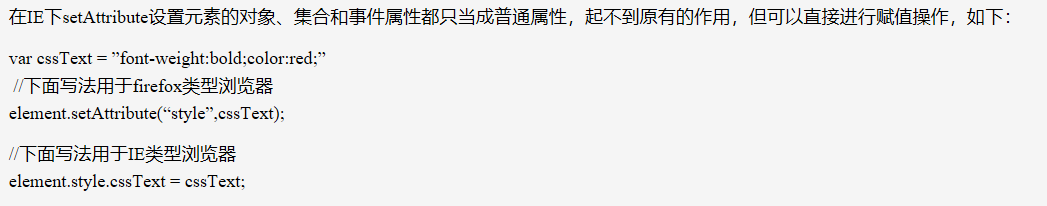
innerText 在火狐中不兼容 要用innerContent代替

document.createElement ie可以传入一段html 其他浏览器传入标签名

setAttribute class与className style与cssText







1. typeof

